Summer 2016 Research with Charlie Curtsinger
PACE: Performance-Aware Computation

Background
Much of my research has focused on how to reliably measure program performance, pinpoint the root causes of performance issues, and provide helpful advice to developers instead of making them comb through thousands or millions of lines of code. I would like to expand these techniques to a broader definition of “performance” and to automatically fix some issues instead of forcing developers to make performance-specific changes to code, which often makes the code less readable and harder to maintain.

The Project
The goal of this project is to build tools to monitor a program as it runs to pinpoint the root causes of high energy usage and poor performance, and to use this information to guide developers to potential improvements or even repair programs automatically. This is a large project that will require many different pieces, many of which would be good summer research projects. Some specific objectives of this project are:

Energy Profiling
A software profiler is a tool that monitors a program’s execution and (usually) reports how much time the program spent executing each function or line of code. I would like to give the same information for energy usage. This will require collecting whole-machine energy use information, then correlating this energy use with the code that was executing during the measurement period.

Load-Adaptive Systems
Systems that experience varying loads, like web servers and games, can reach a point where the load is high enough to bring the entire system to a halt. I am currently working with a student to develop techniques that allow programs to adapt to higher load by decreasing the quality of some approximate computation, rejecting a small fraction of requests, and prioritizing critical parts of the computation. I would like to continue this work by identifying useful techniques for adapting to high load and developing tools to test the resilience of these adaptive systems.

Automatic Performance Repair
Program performance (both in terms of time and energy usage) is sensitive to many different factors, many of which are outside of the programmer’s control. Loading a program into different parts of memory can significantly change its performance (a factor of ten is not unusual), and computer architects are gradually shifting toward systems with many different kinds of processors with different performance and energy demands. We can use information about a program’s runtime and energy use to pinpoint the root cause of a performance problem, then automatically change how the program is loaded into memory or which hardware is responsible for executing different parts of that program in order to reduce energy use while maintaining or improving performance.

Preparation
All of these projects will require significant programming, most likely in C or C++. I do not expect students to know C++, but it should be fairly easy to pick up if you have already seen C and Java. Most of the systems we use for this project will be unfamiliar to you, but you will likely have an easier time learning to them if you have taken 211 or 213. However, these are not strict requirements for working on this project.

I will seriously consider applications from strong students who have not taken 211, 213, or even 207, but some of the above research topics may not be feasible for a student who does not have experience with an object-oriented programming language because of the systems we will need to use in the course of the research project.

This handout and my summer research application are derived from Sam Rebelsky’s summer 2014 research materials. http://www.cs.grinnell.edu/~rebelsky/Department/samr-summer-2014.html
Expectations
I hope to start working with students by the beginning of June, although timing is somewhat flexible. This means you will need to begin your preparations during the spring semester. Once your summer research starts, I expect that you will have invested significant time in learning C++, reading relevant research papers, and familiarizing yourself with the tools we will use for the main research project.

We will meet at least once a week during the summer, but my hope is that you will work independently and with other students to resolve day-to-day technical issues with your project. Solving challenging technical problems with your peers can be a lot of fun, but I will of course be happy to step in to help if you are stuck. That said, I do plan to contribute both ideas and code to the project.

In addition to weekly meetings, I expect you to keep a research journal during the summer. You should take note of your progress toward the end goals for your project, the day-to-day technical issues you ran into, and any surprising results or interesting ideas you find. At the end of the summer and into the fall semester, you will prepare a 5–10 page research paper describing your work in the context of relevant related work. Students working collaboratively on a single research project should work collaboratively on one research paper.

During the fall semester, I will require the student(s) involved in each project to present the results of their summer research. This may be a poster presentation, internal presentation (e.g. CS Extra), or a presentation at an academic conference.

Application Procedures
Applicants must complete my summer research application form and meet with me for a twenty minute interview. During the interview, I may ask you to solve one or two programming problems on the whiteboard in the language of your choice. In addition to your application and interview, I may discuss applicants with colleagues.

I will make final decisions based on a variety of factors including your level of interest in the project, technical preparation, and performance in both CS and non-CS courses. Because women and domestic students of color are under-represented in computer science, I will strive for some equity in the mix of students I select.

This handout and my summer research application are derived from Sam Rebelsky’s summer 2014 research materials. http://www.cs.grinnell.edu/~rebelsky/Department/samr-summer-2014.html
Summer 2016 Research Application
Curtsinger Lab

Name: ____________________________  Major and Year: ______________________

List all of the computer science courses you have taken and the grade you received:

What other courses have you taken that will contribute to your ability to complete this project?

What programming languages do you know, and with what level of proficiency?

What other applicable skills or background would you bring to this project?

Which component of this project do you find interesting, and how would you approach it?
Write a 2–4 paragraph response and include it with your application.

Describe an algorithm that you find particularly elegant, and explain why.
Write a 1–2 paragraph explanation and include it with your application.

Bonus Question: After rolling a pair of six-sided dice 100 times, I observed 31 rolls where the two rolled numbers added to six. Of those 31 rolls, how many times do you think I rolled a two?
Describe your solution to this problem in 1–2 paragraphs and include it with your application.

Please staple all long-answer responses to this sheet before bringing it to me in my office.

This handout and my summer research application are derived from Sam Rebelsky’s summer 2014 research materials. http://www.cs.grinnell.edu/~rebelsky/Department/samr-summer-2014.html